Usage-Based Reading for Inspections of Requirements

Erlansson, Magnus; Thelin, Thomas; Höst, Martin

2002

Usage-Based Reading for Inspections of Software Requirements

Magnus Erlansson, Thomas Thelin and Martin Höst
Dept of Communication System, Software Engineering Group, Lund University, Lund, Sweden
magnuse@telecom.lth.se , thomast@telecom.lth.se , martinh@telecom.lth.se

Abstract

Software inspection has proven to be an effective way to increase the quality of software products. A new reading technique suggested for software inspection, usage-based reading (UBR), has been tested in previous studies, where it showed good defect detection efficiency during inspection of design documents. This study addresses the question whether this is true also for inspections of requirements documents. The idea behind UBR is to let prioritized use-cases direct the reviewer's focus on important parts of the document. Using graduate students as subjects, the UBR approach for inspection of requirement specification was compared with a checklist approach. All defects were classified according to their severity for the function of the final software. The result shows that reviewers using UBR do not find more defects and use more time than those using a checklist. In conclusion, in comparison with a checklist approach, UBR does not make the inspection of requirements specifications more efficient.

1. Introduction

Inspection is a structured method to review software documents and is widely accepted as a cost-effective technique to improve the quality of the software [1, 8]. Code and design specifications have long been exposed to this kind of review [5], but also requirements specifications have been the object of inspection [13]. By inspecting software products early in the development, the defects are not propagated through to later stages of the development where the cost of removal is much higher. About 30 times return on each invested hour of inspection of requirements has been reported [16, 4]. An inspection typically consists of an individual preparation followed by a meeting where the defects are collected and discussed [13]. The reviewers could also look for new defects during this meeting.

This paper mainly focuses on the first step of the inspection process, i.e. the individual defect detection. Reading techniques are used to improve the defect detection in this phase in order to give the reviewers instructions to follow when searching for defects. Some different techniques have been proposed, for example, active design review (ADR) [11], checklist-based reading (CBR) [5] and usage-based reading (UBR) [18]. In this study, the focus is on CBR and UBR. The purpose of the latter technique is to increase the efficiency of an inspection session by using use cases [3] during the defect searching.

The purpose of the study is to evaluate the usefulness of UBR when an inspection of requirement specifications is carried out. The paper compares UBR to CBR in a controlled experiment with student as subjects. The experiment measures the detection of defects with different severity (impact on users) in terms of effectiveness and efficiency.

The main results of the paper are that CBR and UBR is almost equally effective, but CBR is more efficient than UBR. UBR has been under study in three previous experiments, where design inspections were studied [18, 19, 20]. This is the first study using UBR for a requirements inspection.

The paper is outlined as follows. In Section 2, reading techniques are discussed. The method and the results are described in Section 3 and 4. Threats to validity of the experiment and other comments on the results are discussed in Section 5. Finally, there is a conclusion in Section 6.

2. Reading Techniques

A reviewer could use different kinds of reading techniques as aids for defect detection [1]. The aim of reading techniques is to help reviewers to become more effective and efficient in finding defects. This is carried out by giving instructions to the reviewers on what to do during the active reading of a document.

Although many reviewers do not use any formal aids for inspections (i.e. ad hoc), there are several developed approaches that guide the reviewers during the defect detection. The CBR approach is the easiest technique and has long been used by practitioners [8]. More recently, a number of active methods have been developed, as the scenario-based approach suggested by Porter et al., defect-based reading (DBR) [12] and perspective-based reading (PBR) [2]. Furthermore, the reading technique UBR was suggested by Olofsson and

Wennberg [10] and further developed by Thelin et al. [18]. The idea of UBR came from usage-based testing where user scenarios are used to design test cases [15].

The main purpose of UBR is to increase the possibility to find the critical defects from a user's point of view as early as possible during the inspection [18]. This is made by applying prioritized use-cases, which forces the reviewers to focus on parts of the document that is most important for the user. Previous empirical studies have provided evidence for that UBR increases the efficiency as well as effectiveness of design inspections [18, 19]. In particular, Thelin et al. [20] compared UBR to CBR on an inspection of a design document. The result was that reviewers using UBR performed better than reviewers using CBR in design inspections. In this paper, the same research question is addressed for requirements specifications, i.e. which of UBR or CBR is most efficient and effective.

3. Method

Twenty-nine graduate students, following a software engineering course [6] at Lund University, were subjects in the study. The course was optional and most of the students were at their last year in their master education at the University. Two treatments were chosen for the study, Usage-based reading (UBR) and checklist-based reading (CBR) used as a baseline. The students first assigned themselves to six groups with about five people in each. As a first step of the inspection experiment the students participated in a start-up meeting. At this meeting they got their individual package of documents to be inspected and forms to fill in. They also got detailed instructions for the remaining steps in the process. Before the inspection, which was the second step in the process, the two treatments (CBR approach or UBR approach) was randomly assigned to three groups each. 14 students were assigned to use CBR during inspection and 15 were assigned to use UBR. Then the students inspected the requirement specification individually at home using one of the two chosen reading techniques. They could use maximally about two hours. As a third step, each group conducted a meeting where they discussed the defects found and produced a common list of defects that they could agree on.

The inspected document was a requirement specification for a taxi management system (10 pages) including a short overview of the system. The requirements were written in a feature style notation [9] using natural language (English) and the document included a glossary, a state chart, and a context diagram. Before the experiment 14 defects were seeded into the requirement document, in which there could be a number of other defects. The seeded defects were mostly defects that previously had been removed from earlier versions of the requirements specification. The checklist [17] was very short and asked the reviewers to check the document for correctness, completeness and consistency as well as that the requirements were unambiguous, realistic and verifiable. The use-case document (10 pages) consisted of 24 use-cases in prioritized order with the most important use-case first. These use-cases were used earlier by Thelin el al [19] in a UBR-experiment where a design document was inspected. All documents and forms were written in English. No students have English as their native language.

After the inspection the students were asked to grade their previous experience (5 areas with score 1 to 5 each) and reveal their educational background. The answers were used during the interpretation of the result of the study. For ethical reasons the students did not write their names on any form. Instead each group got unique identification numbers to distribute within the group in order to couple a form with a group and a treatment.

As a preparation step before the analysis of the result, the defects found by the reviewers were discussed and classified. First, a list of all defects found individually was produced. If there were more than one defect referring to the same original defect in the document they were regarded as one defect only. In case the reported defects were not judged as true defects (after precise consideration by the authors of this paper), they were discarded. Then all defects in the final list were classified (by the authors of this paper) according to their severity for the function of the final software product. The following classes were used.

- **A-defects**: The most severe defects (in critical functions or frequently used functions. These defects were also considered most important for the user.
- **B-defects**: Moderately severe defects. These defects were found in not so important functions or in not frequently used functions.
- **C-defect**s: Minor defects that do not affect the function very much.

For the statistical comparison of the results, a nonparametric test (Mann-Whitney U-test) was used. P-values less than 0.05 were regarded as significant.

4. Results

The results from the investigation of the students' previous experience and educational background did not reveal any differences between the reviewers using CBR compared with those using UBR as aid during the inspection (see Table 1 and Figure 1).

Educational Background	CBR	UBR
Computer or Electrical engineering (D or E program in Sweden)	7 students	6 students
Industrial/economic or surveying engineering (I or L program in Sweden)	6 students	8 students

Table 1. The educational background of the students acting as reviewers in the experiment. Two students (one from each group) did not answer (optional by ethical reasons) the questionnaire. Among the replying students there was no significant difference in educational background between the CBR group and the UBR group.

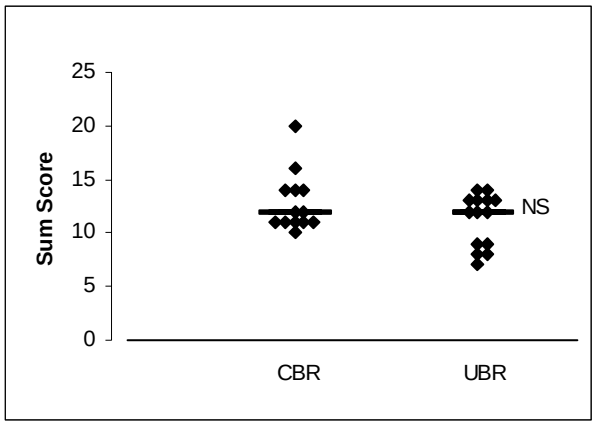

Figure 1. Previous experience of the students that acted as reviewers in the experiment. The dots represent the individual sum of scores within five experience areas (programming, requirement inspections, use-cases, developing taxi system and using taxi). The horizontal lines represent the medium values. There was no significant difference between the groups.

A first list of 89 possible defects reported individually from the 29 reviewers was produced. At the first analysis of this list the number of defects were reduced dependent on one of the following reasons.

- Just minor language remark/ criticism and therefore discarded. (1 case)
- Duplicates of other defects. (11 cases)
- Conflicts only with use-cases and not within the requirement document and therefore discarded. (5 cases)
- Not considered as real defect (after careful judgment by the authors of this paper) and discarded. (25 cases)

47 true defects were remaining and they were classified in 16 A-defects, 15 B-defects and 16 C-defects.

At an inspection of a document, effectiveness refers to the number of defects found independent of the time consumed by the reviewers. Defects that could be most relevant for the user of the final software product were assumed to be A-defects or maybe A- and B-defects together. Figure 2 shows the effectiveness of the individual inspection considering the different classes of defects and comparing the two different reading techniques, CBR and UBR. There were no significant differences except for B-defects and C-defects. Thus, UBR-reviewers were significantly more efficient than CBR-reviewers in finding B-defects but less effective in finding C-defects. No differences could be seen in their effectiveness in finding A-defects or in finding both A- and B-defects or in finding all defects (A+B+C).

The efficiency during defect detection refers to the number of defects found per time unit. Since reviewers consumed different amount of time for their individual inspection, the result differs from the effectiveness result. Figure 3 (below) shows the efficiency in terms of number of defects found per hour. The result shows that the CBR approach was significantly more efficient in finding defects than the UBR approach. This was seen for A- and B-defects as well as all defects (A+B+C). UBR-reviewers were significantly more efficient in finding B-defects. Interestingly, most UBR-reviewers found no C-defects at all.

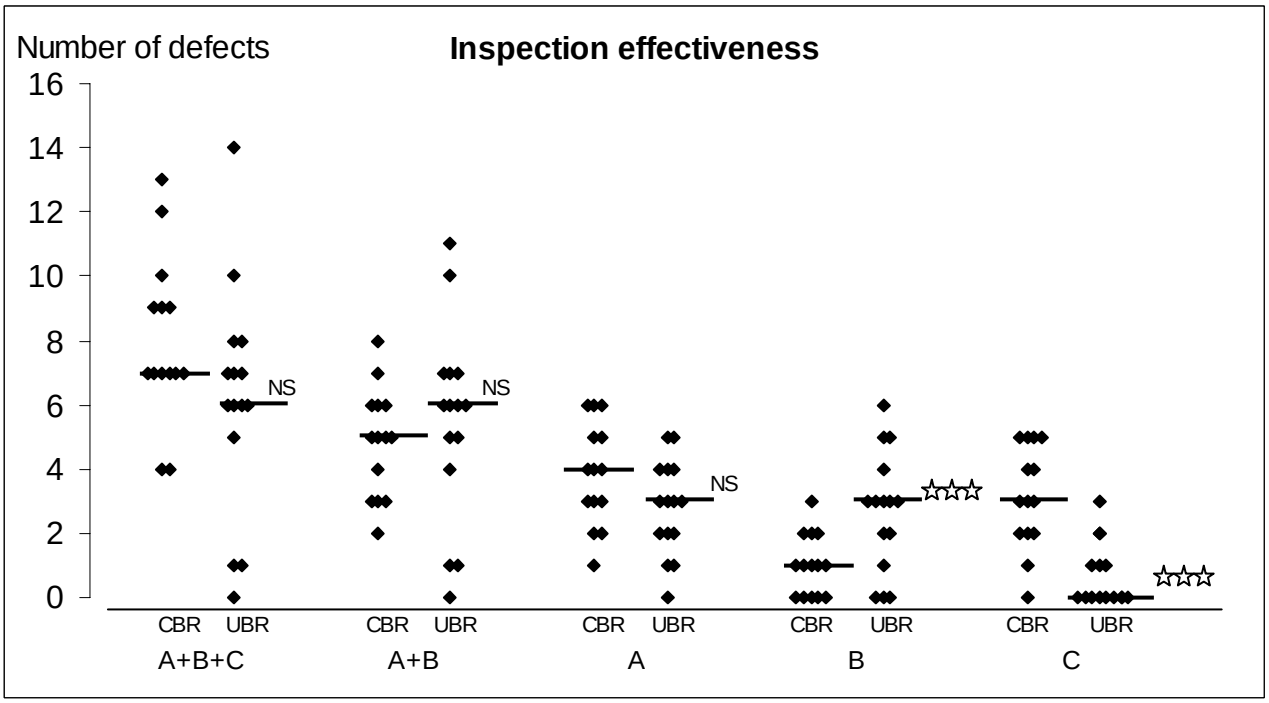

Figure 2. Individual inspection effectiveness. The defects are classified as severe (A), moderately severe (B) and not severe (C) defects and the individual results are shown as black squares. Horizontal lines represent medium values. There were 14 reviewers using the checklist-based reading (CBR) and 15 reviewers using usage-based reading (UBR). (NS= non-significant, * means p<0.05, ** means p <0.01 and *** means p<0.001)

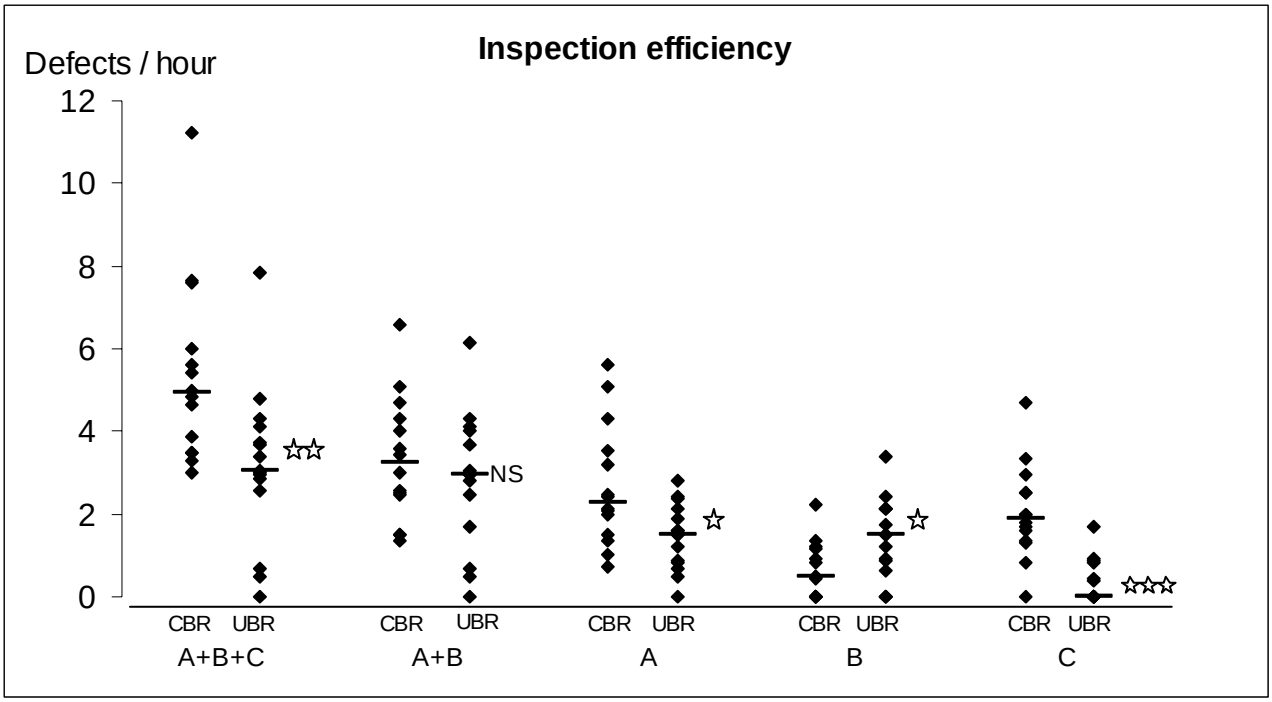

Figure 3. Individual inspection efficiency. The defects are classified in severe (A), moderately severe (B) and not severe (C) defects. Individual number of defects found per hour is shown as black squares. Horizontal lines represent the medium value. There were 14 reviewers using the checklist-based reading (CBR) and 15 reviewers using usage-based reading (UBR). (NS= non-significant, * means p<0.05, ** means p <0.01 and *** means p<0.001)

Apart from classifying the defects in A, B and C defects according to their impact on the user, all defects were also categorized in different kinds of defects (see Figure 4). The results indicated that the most severe defects were either missing information or inconsistent information.

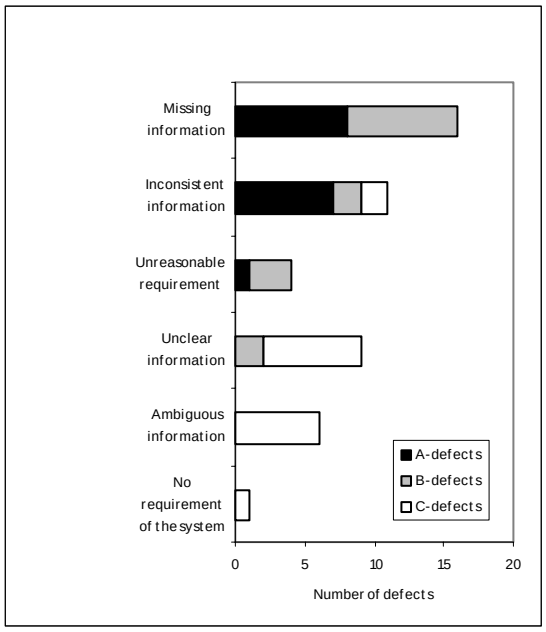

Figure 4. The distribution of defects according to category (6 different) and severity (A, B and C). A-defects were considered most important for the user

5 Discussion

In this study, we show that usage-based reading is not better than conventional checklist-based reading for inspections of software requirement documents. This is shown both for the inspection effectiveness (total number of found defects) and inspection efficiency (number of found defects per time unit).

5.1 Threats to the validity of the study

The following conditions are identified as possible threats to the internal validity of the study.

- Individual inspection conducted at home (non-controlled). The validity is dependent on that the students honestly report the actual time consumption for the inspection. On the other hand it is assumed that most individual inspections in the industry are conducted very informally, similar to the conditions in the present study.
- Inspected document and written instructions in English and not in the reviewers mother tongue. It could have been a burden for some of the students. Though, they could use their own mother tongue when they described the defects in the form. Many Swedish companies write their software documents in English.
- No training period for the students before the real inspection. They might have been too inexperienced to make a real inspection especially using UBR as a reading technique.
- Not completely random assignment on an individual basis, since the students form the inspection groups before the random assignment of the treatments to the groups. However, the previous experience and educational background turned out to be quite similar in both treatment groups (se Figure 1).
- The classification of defects according to their severity is very subjective. However, defects were discussed by all the authors of this paper before the final decision.

Whether the findings could be generalized to be valid for requirements inspections also in industry is dependent of the external validity. The following condition is identified as possible threat to the external validity of the study.

- The use of students as reviewers. One can argue that the student's experience is too low compared with practitioners in the industry. However, Porter and Votta [14] reported that although students might have lower performance the result of an experiment with students will be the same as if professionals in the industry have been subjects in the study.

5.2 Comments on the results

Considering that the purpose of UBR is to get the reviewers better focused on the most important parts of the document, it is not surprising that the UBR-reviewers revealed much less C-defects than the CBR-reviewers. Actually, most of the UBR-reviewers found no C-defects at all and apparently did not waste time on defects that have low impact on the end user of the software. Furthermore, the B-defects with moderate impact on the end user were detected more easily by the UBR-

reviewers. However, the defects that are supposed to have the greatest impact on the user, the A-defects, were best detected by CBR-reviewers. One explanation could be that the UBR approach was too heavy and exacting for the reviewers. Most of the A-defects were missed requirements or conflicts within the requirements document (see Figure 4). To detect these kinds of defects the reviewers may need to compare relatively large parts of the document. It may be easy when the reviewer just need to read the requirement document. UBR-reviewers, however, need to read also the use-cases (10 pages in this study) and may have been overloaded with information. Kelly et al [7] found that an increase in the number of pages to be inspected lead to a drop in the number of defects found by the reviewers. A possible indication of an overload in our experiment was that the UBR-reviewers consumed much more time during the inspection than the CBR-reviewers. Further studies may elucidate this topic.

The aim of UBR is to help the reviewers to focus on the users' need during inspection. A common instrument for that purpose is use-cases, which in the case of UBR are prioritized and sorted in order dependent on the importance for the user. For inspection of design documents Thelin et al speculated about that it might be effective to develop use-cases on the fly during inspection instead of utilizing pre-developed use-cases. Thus, they conducted an experiment where one group of reviewers got pre-developed use-cases and another group got only the purpose of each use-case and therefore had to develop more detailed use-cases as a part of the inspection. However, the results showed that it was better to include complete use-cases from start.

One can also speculate on whether other techniques could be used to focus on the users' need. One suggestion could be that a group of users (or people close to users) prioritize requirements or parts of the requirements specifications. Similar to prioritized use-cases, such a prioritization directly marked in the requirement should help the reviewers to find defects that have most impact on the user.

Opposite to our result in this study on requirements inspection, Thelin et al. [20] found that UBR was better than CBR for inspection of design documents. The experimental design was very similar to the design of our study. Thus, students were used as subjects and the defects were classified according to the severity and impact to the end user. Also, the inspected document domain (a taxi management system) and the use case document were the same. However, one difference might have influenced the result. In the previous study with the design document, the students were trained prior to the experiment. They had an introductory lecture about the taxi system and were then introduced to the reading techniques by trying the UBR/CBR on a smaller system. Our study did not include a training phase before the experiment.

To summarize, this experiment shows that UBR may not be appropriate to use on requirements inspections. This could either be because a checklist is good enough to use for such an inspection or because the reviewers need more training before utilizing the UBR approach. Further replications of the experiment should get a deeper knowledge of what causes the outcome.

6. Conclusion

This study has compared usage-based reading (UBR) and checklist-based reading (CBR) for inspection of software requirements using students as reviewers. From this study, in which we have compared the ability of reviewers to find the defects with the most severe impact on the end user, we conclude that

- the effectiveness (total number of severe defects found) of requirement inspection was almost the same for reviewers using UBR and CBR as inspection aid.
- the efficiency (number of severe defects found per inspecting time unit) of requirement inspection was higher for CBR than for UBR approach.

7. Acknowledgement

This work was partly funded by The Swedish National Agency for Innovation Systems (VINNOVA), under a grant for the Center for Applied Software Research at Lund University (LUCAS).

8. References

[1] A. Aurum, H. Petersson and C. Wohlin, "State-of-the-Art: Software Inspections after 25 Years", *Software Testing, Verification and Reliability,*12(3), 2002, pp. 133-154.

[2] V.R. Basili, S. Green, O. Laitenberger, F. Lanubile, F. Shull, S. Sørumgård and M.V. Zelkowitz, "The Empirical Investigation of Perspective-Based-Reading", *Empirical Software Engineering: An International Journal*, 1(2), 1996, pp. 133-164.

[3] A. Cockburn, *Writing Effective Use Case*s, Addison-Wesley, USA, 2001

[4] E.P. Doolan, "Experience with Fagan's Inspection Method", *Software-Practice and Experience* 22(2), 1992, pp.173-182.

[5] M.E. Fagan, "Design and Code Inspections to Reduce Errors in Program Development", *IBM Systems Journal*, 15(3), 1976, pp.182-211.

[6] M. Höst, "Introducing Empirical Software Engineering Methods in Education", in *Proceedings of Conference on Software Engineering Education and Training*, 2002, pp. 170-179.

[7] J.C. Kelly, J.S. Sherif and J. Hops, "An Analysis of Defect Densities Found During Software Inspection", *Journal of Systems Software*, 17, 1992, pp. 111-117.

[8] O. Laitenberger. and J.M. DeBaud, "An Encompassing Life Cycle Centric Survey of Software Inspection", *Journal of Systems and Software*, 50(1), 2000, pp. 5-31.

[9] S. Laueson, "Software requirement – Styles and Techniques" Addison-Wesley, Pearson Education Limited, London , 2002.

[10] M. Olofsson, and M. Wennberg, "Statistical Usage Inspection", Master's Thesis, Department of Communication Systems, Lund University, CODEN:LUTEDX(TETS-5244)1-81/1996&local. 9, 1996.

[11] D.L. Parnas and D.M. Weiss, "Active Design Reviews: Principles and Practices", *Proc. of the 8th International Conference on Software Engineering*, 1985, pp. 418-426.

[12] A. Porter, L. Votta and V.R. Basili, "Comparing Detection Methods for Software Requirements Inspections: A Replicated Experiment, *IEEE Transactions on Software Engineering*, 21(6), 1995, pp. 563-575.

[13] A. Porter, H.P. Siy and L. Votta, "A Survey of Software Inspections", *Advances in Computers*, 42, 1996, pp.40-77.

[14] A. Porter and L. Votta, "Comparing Detection Methods for Software Requirements Inspection: A Replication Using Professional Subjects", *Empirical Software Engineering*, 3(4), 1998, pp. 355-380.

[15] P. Runeson and C. Wohlin, "Statistical Usage-Testing for Software Reliability Control", *Informatica* 19(2), 1995, pp. 195-207.

[16] G.W. Russel, "Experience with Inspection in Ultralarge-Scale Developments", *IEEE Software* 8(1), 1991, pp. 25-31.

[17] I. Sommerville, "Software Engineering", 6th edition, Addison-Wesley, Pearson Education Limited, London, 2001.

[18] T. Thelin, P. Runeson and B. Regnell, "Usage-Based Reading – an Experiment to guide reviewers with use-cases", *Information and Software Technology*, 43(15), 2001, pp. 925-938.

[19] T. Thelin, P. Runeson, C. Wohlin, T. Olsson and C. Andersson, "How much Information is Needed for Usage-Based Reading? – A series of Experiments", to appear in: *International Symposium on Empirical Engineering*, 2002.

[20] T. Thelin, P. Runeson and C. Wohlin, "An Experimental Comparison of Usage-Based and Checklist-Based Reading", Submitted to: *IEEE Transactions on Software Engineering*, 2002